The Artificial Intelligence and Psychology Project

Departments of
Computer Science and Psychology
Carnegie Mellon University

Learning Research and Development Center
University of Pittsburgh

2

MODELLING HUMAN COGNITIVE DEVELOPMENT WITH EXPLANATION-BASED LEARNING IN SOAR

Technical Report AIP - 120

Tony Simon
Department of Psychology

Carnegie Mellon University
Pittsburgh, PA 15213 U.S.A.

February 2, 1990

This is the final report on the research supported by the Personnel and Training Research Program, Psychology Sciences Division, Office of Naval Research, under contract N00014-86-K-0349.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; Distribution unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AIP - 120	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Carnegie Mellon University; 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Computer Sciences Division Office of Naval Research (Code 1133)
6c. ADDRESS (City, State, and ZIP Code) Department of Psychology Pittsburgh, PA 15213	7b. ADDRESS (City, State, and ZIP Code) 800 N. Quincy Street Arlington, VA 22217-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Same as Monitoring Organization; 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N00014-86-K-0678
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS p40005ub201/7-4-86

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
N/A	N/A	N/A	N/A

11. TITLE (Include Security Classification)

Modelling human cognitive development with explanation-bsed learning in Soar

12. PERSONAL AUTHOR(S) Tony Simon

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Technical	FROM 86Sept15 TO 91Sept14	2Feb1990	12

16. SUPPLEMENTARY NOTATION

17. COSATI CODES		
FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)

cognitive development
Soar
explanation-based learning

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

SEE REVERSE SIDE

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code)	22c. OFFICE SYMBOL
Dr. Alan L. Meyrowitz	(202) 696-4302	N00014

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

Unclassified

Explanation-based learning has been shown to be an effective method for operationalising concepts implicit in a problem solver's knowledge base. Demonstrations have thus far used mainly deductive techniques over complete domain theories and with respect to limited tasks. This paper outlines some early work on augmenting EBL with a simple inductive capability and applying it to the real world domain of modelling the development of Piagetian number conservation concepts in children.

MODELLING HUMAN COGNITIVE DEVELOPMENT WITH EXPLANATION-BASED LEARNING IN SOAR

Tony Simon

Department of Psychology
Carnegie Mellon University
Pittsburgh PA 15213
USA
tonys@cs.cmu.edu

Abstract

Explanation-based learning has been shown to be an effective method for operationalising concepts implicit in a problem solver's knowledge base. Demonstrations have thus far used mainly deductive techniques over complete domain theories and with respect to limited tasks. This paper outlines some early work on augmenting EBL with a simple inductive capability and applying it to the real world domain of modelling the development of Piagetian number conservation concepts in children.

1. Introduction

The last few years has seen a profusion of work in Explanation-Based learning (EBL) [1, 9]. The task of an EBL system is to accept a training instance and show that it is or is not a member of a given concept, thereby automating the classification process for future instances. This is done by problem solving over a domain theory which supports a mapping between the predicates of the training instance and those of the concept definition. Resulting generalisations must comply with an operationality criterion which limits the operationalised concept description to one which is easily evaluated for new instances.

Most EBL systems have been developed using either very limited tasks such, as stacking blocks [9], or limited approximations of real world tasks such as learning concepts that define suicide or kidnappings [1]. This approach enabled researchers to supply fixed domain theories which were deductively complete, at least for a limited range of examples. As a result, the applicability of any EBL system was limited and so a current concern is to augment EBL with inductive capabilities so that it is possible to work in more complex and unpredictable domains [13].

In this paper I shall outline the early stages of work to extend the use of EBL in a novel direction. The EBL approach is applied to the modelling of a domain that is both a real world learning problem and one which has been extensively studied by the field of developmental psychology. This is the formation of Piagetian number conservation concepts in children. In order to model the development of these concepts the EBL method is adapted by replacing a complete domain theory with a very minimal one. Along with that minimal domain theory is supplied the capability for induction by simple causal attribution. The claim is that the combination of EBL with a novel performance model of quantity conservation development results in one of the first simulations of this heavily researched, yet ill-explained, developmental phenomenon.

2. A Model of Conservation Development

Work on Piagetian conservation concepts was for a long time a major preoccupation for the field of cognitive developmental psychology. This is mainly because one of the central tenets of Piagetian theory [11] is that the acquisition of conservation concepts is a critical milestone in the child's development of mature conceptual capabilities. Conservation concepts can be defined as the understanding that, in the face of certain irrelevant (though often misleading) transformations - such as spreading a row of objects - some aspects of those objects - such as numerical value - remain invariant.

In a typical experiment on number conservation a child is shown two rows of objects generally with the same number of objects in each row. The objects are usually lined up in 1-to-1 correspondence. The child is asked if the two rows have the same number of items in them. Once this is agreed, one of the rows is transformed (usually by lengthening or shortening) and the child is asked the "conservation question"; whether the rows still have the same number of items in them. Children who consistently answer that the two rows do have the same number, and can explain why, are said to have acquired the conservation concept with respect to number. Those who are misled by the perceptual features and say that the longer row has more objects are classified as non-conservers.

The critical shift in the development of conservation concepts is for the child to stop attending to the details of the objects themselves and to focus on the transformation that was applied to them. Thus, what is required is the learning of the class of operations that affect a given dimension (such as quantity) and the class of operations that do not affect it. In this way the child can infer the effect on quantity simply by observing the action applied to the materials. Once a concept such as "quantity-preserving transformation" has been learned, it can be applied to objects of any quantity to assess conservation. Many experiments have been carried out on a wide range of conservation concepts for number, mass, area, volume and even existence [3]. Concentrating on conservation of number and other kinds of quantity one can observe that a set of regularities exist in the literature. By a regularity we mean a finding that is consistently reported and for which there is little or no disconfirming evidence. The four main regularities are presented below.

1. Young children in the 3-6 year-old age range are able to obtain specific quantitative values for small sets (e.g. up to about 5) of objects e.g. [2].
2. Young children in the 3-6 year-old age range are unable to obtain specific quantitative values for larger sets of objects e.g. [4].
3. Children who have not acquired the conservation concept nevertheless can still correctly answer the conservation question when they can obtain a specific quantitative value for objects concerned e.g. [15].
4. Children who have not acquired the conservation concept cannot correctly answer the conservation question when they cannot obtain a specific quantitative value for objects concerned e.g. [5].

To explain these regularities I have formulated the following model to account for the development of quantity conservation; I shall call it the Quantification model (or Q-model). The Q-model says that children learn quantity conservation by generalising the effects of transformations on countable arrays into the classes of quantity-preserving and quantity-modifying transformations. This comes about in the following way. The child computes a specific value for a given set of objects and then recomputes a value for them after some kind of transformation has occurred (such as building a tower with a pile of blocks or setting a table with silverware from a drawer). Then, if the two values match, a plausible inference can be made that the quantity was not affected by the transformation concerned. The Q-model assumes that the child forms conservation concepts by "explaining" such outcomes in terms of the actions and by generalising over the specific objects and amounts involved in the transformations. Eventually a transformation, such as spreading, will be represented without any reference to specific objects or agents and it will always be expected to have a quantity-preserving effect [6]. According to the Q-model, it is through a process of explanation-based learning that children acquire conservation concepts that enable them to infer certain outcomes by attending to the transformation alone. So, just is in other EBL models, here the quantitative effect of operationalised actions can be easily determined, even for material where a specific quantitative value cannot be directly computed.

Further support for this model comes in the form of two more regularities that can be observed in the conservation literature.

5. Correct answers to conservation of quantity are achieved on tests of discrete stimuli (e.g. buttons, coins, shapes) before tests of continuous stimuli (e.g. liquids) e.g. [15].
6. Training conservation on discrete quantities alone transfers to continuous quantities once the conservation concept has been formed e.g. [17].

Both of the above regularities provide evidence for the claim that the learning and generalisation of the effects of various transformations takes place only in cases where the materials are countable[1]. Then, once the generalised conservation concept has been learned, it transfers to non-countable materials.

3. Modelling Development with Explanation-Based Learning

In the rest of this paper I will show that combining a slightly modified EBL approach with the Q-model provides support for this theory of conceptual development. The simulation model that I shall describe here, called ABC.Soar for Analysis-Based Concept formation, is being developed using the Soar architecture [7]. I shall briefly review the mapping of EBL into Soar as described by Rosenbloom and Laird [12] before

[1] I use the term counting to include "subitizing", a capability to perceptually apprehend small numbers.

providing an abstracted description of the initial stages of the formation of quantity conservation concepts by an early version of ABC.Soar.

Rosenbloom and Laird [12] have shown how EBL maps directly into the Soar architecture [7]. Using the "Safe-to-Stack?" example [9] they describe a Soar system that has an operator, "Safe?(x,y)", which examines information about a training instance and can be implemented when it can compute if it is safe to stack x on y. Initially, the instance and concept definition representations are not stated in comparable terms. Thus the operator cannot evaluate the instance in terms of the concept definition and so it fails to apply. This results in the creation of a subgoal. Rosenbloom and Laird show that the domain theory is a problem space that can be selected for that subgoal and which contains the required operators to translate the goal concept definition into those predicates that describe the instance. Once that has been done the "Safe?" operator can be applied to compute whether the instance fits the definition or not. The chunk that is created by the successful termination of that subgoal then describes the newly operationalised concept definition. This is stated in terms only of the critical predicates that were required in the computation performed by the "Safe?" operator and which depend on those that existed before the subgoal was created.

Since the point of this paper is to illustrate the application of EBL to modelling the formation of conservation concepts, I will only briefly discuss the other aspects of the model that are not directly involved with this process. The top goal of ABC.Soar is to observe a transformation to a set of objects (R2 below) and to return a judgement about whether the number of objects is the same as it was before the transformation took place. Below is a schematic representation of the task.

```
         Time1            Transformation           Time2

R1    0 0 0 0 0                                  0 0 0 0 0
R2    0 0 0 0 0 ==>      "SPREADING"    ==>    0   0   0   0   0
```

ABC.Soar splits into a performance and an acquisition component as is common for EBL systems [1]. The top problem space implements the performance component. There is an operator to observe and represent the transformations that are carried out and a response operator that returns a judgement about the numerical effect of transformations. A comprehension operator is applied to each new input. This applies knowledge that the system has about such an input so that it can be understood and acted upon. If not enough is known about the transformation to produce a conservation judgement then the Comprehend operator will fail to apply and will create an impasse. In the resulting subgoal the acquisition component works to augment the problem solver's knowledge to the extent that the Comprehend operator can interpret the input and the response operator can state the inferred effect of the transformation.

Using the task represented above, we shall assume that the Comprehend operator has created such an impasse. The comprehend problem space will be selected in order to try to implement the Comprehend operator. Given sufficient knowledge, the Categorise-input operator will be selected to classify the transformation (in this case as an instance of a concept that we will call "spreading-actions"[2]). Once this has been done the Same-number? operator should be able to infer what the effect of such an action is by access to the classes of transformations (or conservation concepts) that it has already learned. In either case, should the operator fail to apply due to a lack of knowledge it will create a further impasse. At that point, ABC.Soar will try to define or revise concepts that either merely describe classes of actions, or ones that capture the numeric effects of those actions.

There is not the space here to address the formation of action concepts and it is dealt with in detail elsewhere [16]. Instead we shall continue the example from the point where the action has been categorised as an instance of the concept "spreading-actions". Having categorised the transformation the Same-number? operator is selected to produce an answer to the conservation question. However, the problem solver does not know the expected numerical effect of the spreading transformation and so the operator fails to apply. The impasse that results signals the initiation of the process of creating an explanation for whether this transformation is an instance of the "quantity-preserving" or "quantity-modifying" class of actions.

---

[2]Internally, the system uses gensyms in place of such names. Concepts are named here for ease of reading.

Just as in the mapping illustrated by Rosenbloom & Laird [12], the domain theory for this task will be the set of operators in the domain-theory problem space that is selected for this subgoal. An explanation of the conservation concept is the set of those operators that fire to implement the Same-number? operator. The operationality criterion relates to the fact that those operators will have to augment the representation on the state with predicates in which the implementation productions for the Same-number? operator are stated.

3.1. Domain Theory

The domain theory that is provided for ABC.Soar is extremely simple. It consists only of four operators. I shall state them below in a simplified pseudo-code form and instantiate them with some example values.

```
Quantify (v)                                                          1
transform(t,r2) /\ ~number-time2(r1,v1) /\ ~number-time2(r2,v2)
         -> number-time2(r1,v1) /\ number-time2(r2,v2)

Changed? (r,v)                                                         2
number-time2(r1,x) /\ number-time2(r2,x) /\ name(x,5)                      a
         -> unchanged(r2,v)
number-time2(r1,x) /\ number-time2(r2,y) /\ name(x,5)                      b
         /\ name(y,6) -> changed(r2,v)

Link (t,v)                                                            3
transform(t,r) /\ number-time2(r,v2) /\ ~link(t,v2)
        -> link(t,v2)

Effect? (t,v2)                                                        4
transform(t,r) /\ link(t,v2) /\ unchanged(r,v2)                            a
        -> Same-number(v2,true)
transform(t,r) /\ link(t,v2) /\ changed(r,v2)                              b
        -> Same-number(v2,false)
```

Operator #1 enables the problem solver to count both rows of up to five objects after the transformation[3]. The capability to quantify only up to five objects is justified by the regularities described above. Operator #2 is implemented by one of two productions depending on whether the values of the two rows are the same or not. Operator #3 is a

[3]This operator is a gross oversimplification of the quantification process but is sufficient to serve the purpose of a demonstration in this example system.

causal attribution operator very like those employed by Lewis [8] for creating new procedures in Human-Computer interaction tasks. It also reflects the sophisticated causal reasoning abilities of young children [14]. This operator fulfills the inductive role of attributing changes in the materials, such as those relating to number (as in the example above) and length, to actions on those same materials. Previously those actions would have carried no representation of their effects on such attributes. Operator #4 implements the Same-number? operator in terms of the causally attributed numerical effects of transformations. Operators #1 & #3 are default operators. This means that, even if applicable, they will only fire if no other operators do. In this way there is a simple means of building up the domain theory using a weak method. However, if any learned knowledge is applicable, the operators that it recommends will be applied in preference to the default operators.

3.2. Conservation Concept Formation

Recall that ABC.Soar has categorised the transformation as "spreading" but that the Same-number? operator has failed to apply and created the subgoal in which the domain-theory problem space has been selected. What follows is an abstracted trace of the formation of the conservation concept for the task represented above[4]. Having observed the spreading action, the outcome that the transformed row is now longer than the untouched row has been added to the representation. The goal for the child is to answer whether there is still the same number of objects in the two rows as a result of the transformation. This goal will have the effect that concepts formed concerning the effects of transformations in this activity will be organised around their numerical outcomes [10].

The representation that is encoded by ABC.Soar is presented here in a very simplified manner but one that is sufficient for the purposes of this paper. Below is a representation of the state after observing and categorising the transformation.

```
Name(a,row1) Name(b,row2) Colour(a,red) Colour(b,blue)
```

[4]The reader should note that the difference between this process and the standard EBL case is that the problem solver has no pre-existing abstract description of the target concept of, for example, "quantity-preserving transformation".

```
Objects(a,square)          Objects(b,circle)
Transform(t,b)             Name(t,Spreading)
Number-Time1(a,x)          Number-Time1(b,x)
Length-Time1(b,y)          Length-Time2(b,z)
Name(x,5)         Name(y,10)        Name(z,15)

Link(x,y)
 transform(t,r) /\ length-time2(r,15) -> link(t,15)                      5
Quantify(x)
 transform(t,r2) -> number-time2(r1,5) /\  number-time2(r2,5)            6
Link(x,y)
 transform(t,r) /\ number-time2(r,5) -> link(t,5)                        7
Changed?(r,v)
 number-time2(r1,5) /\ number-time2(r2,5) -> unchanged(r,5)              8
Effect?(t,5)                                                           9
  transform(t,r) /\ link(t,5) /\ unchanged(r,5)
        -> same-number(5,true)
```

The trace above describes the problem solving in the domain theory space which creates an initial definition for the concept of "quantity-preserving transformation". Production #5 links the length outcome to the transformation but that does not enable the Same-number? operator to fire because no numerical effect was established. (The Changed? operator firing for this outcome is not shown). The only applicable operator now is Quantify and its application results in a numeric outcome for each row. This enables a new link to be created (productions #6-7) for the row that was transformed. The Changed? operator now applies and computes an unchanged numeric value for the transformed row by comparing its number to the untouched row (production #8). Now the state has a sufficient representation to make the Effect? operator applicable. It applies and produces the result that the transformation did not cause a change in the quantity of the objects (production #9). This terminates the subgoal because the same-number predicate implements the Same-number? superoperator. As is always the case in Soar, successful termination of a subgoal causes a chunk to be built to summarise the processing in the subgoal. Soar's backtracing process for building chunks not only determines the explanation structure but it also defines the conditions for the new rule by computing dependency to those predicates that existed before the impasse was created [12]. In this case the transformation and the final values of the compared rows are the critical elements. So the result is an initial operational definition of a quantity conservation concept stated in the form of the rule:

```
Same-Number?(r,5)
transform(t,r2) /\ number-time2(r1,5) /\ number-time2(r2,5)
        -> same-number(5,true)

Where t = "spreading-actions"
```

Having returned this result the response operator can apply and report that the number was unaffected by the spreading transformation. This satisfies the top goal of ABC.Soar. However, the problem solver is still far from performing like a perfect conserver. It has built two concepts, both of which are over-specific. The "spreading-actions" concept is stated only in terms of the details of the initial instance. The conservation concept states only that spreading causes conservation for collections of five objects. Yet, these concepts will be used and generalised. Future instances of spreading countable rows of different numbers will cause the above conservation concept to be variabilised. Each time spreading is seen, conservation concepts with different values will be recalled. These will be generalised, leading to a rule which states that spreading any number of things leads to no change in that number. So, as it encounters difficulties like the inability to categorise new instances or the making incorrect inferences due to incomplete concept definitions, ABC.Soar will impasse into its acquisition component. Here it can deliberate over the problem at hand and build chunks for the solutions it finds. Thus, ABC.Soar will exhibit a graceful integration of deliberate and automatic processing which is mediated by the interaction of its knowledge and the problems that are encountered. Expertise acquisition will be gradual and experience-based, just as any plausible model of cognitive development should be.

4. Discussion

This paper has illustrated how the EBL approach can be used to model a real-world example of conceptual development in children. The main modification to the standard EBL technique was in terms of the domain theory. In ABC.Soar this is cast as one simple domain-specific capability - quantification - and a general weak method - causal attribution - for linking outcomes to actions in the task domain. This enables a sort of bootstrapping of the domain theory that can be augmented through application to subsequent input. If the knowledge that has been acquired is adequate then the problem solver can proceed relatively effortlessly. However, whenever difficulties are encountered, ABC.Soar will create impasses leading to the definition of new concepts or the revision of existing ones.

The current implementation of ABC.Soar was a simplified experiment to test the feasibility of this approach. As such it is incomplete in a number of senses. One aspect is the organisation and accessing of concepts. As ABC.Soar's knowledge grows, a more sophisticated access mechanism will be required to enable the system to recall relevant concepts and reject ones that appear irrelevant. Cued recall in Soar's datachunking mechanism provides a candidate solution. Related issues are the revision and organisation of concepts which both require further research. Of course ABC.soar cannot be considered a complete model of the development of conservation concepts until it can also account for the development of the quantification capability that it currently employs as part of the minimal domain theory.

Nevertheless the demonstration reported here is encouraging in two respects. First, the construction of ABC.Soar suggests that, without much modification it is possible to extend the EBL approach to real world domains. Second, the combination of a novel performance theory for a large body of psychological data and EBL provides support for a theory of conceptual development which is a significant departure from previous models. Conservation development is often explained in terms of the learning about the reversibility of operations; where the child comes to represent the fact that transformations can be reversed and the original state of affairs will be restored. It is hoped that ABC.Soar will inspire the development of other new models by applying machine learning techniques to phenomena in human cognitive development where they can be of great assistance in the much needed process of formalisation.

This work was supported by the Personnel and Training Research Program, Psychological Sciences Division, Office of Naval Research under contract number N00014-86K-0349.

References

1. DeJong, G. & Mooney, R. "Explanation-based learning: An alternative view". *Machine Learning 1* (1986), 145-176.

2. Frye, D., Braisby, N., Lowe, J., Maroudas, C. & Nicholls, J. "Young children's understanding of counting and cardinality". *Child Development 60* (1989), 1158-1171.

3. Gelman, R. & Baillargeon, R. A review of some Piagetian concepts. In *Handbook of Child Psychology*, Wiley, New York, 1983.

4. Gelman, R. & Gallistel, R.C.. *The Child's Understanding of Number.* Harvard University Press, Cambridge, Mass, 1978.

5. Halford, G.S. & Boyle, F.M.. "Do young children understand conservation of number?". *Child Development 56* (1985), ??.

6. Klahr, D. Transition processes in quantitative development. In R.J. Sternberg, Ed., *Mechanisms of Cognitive Development*, Freeman, New York, 1984.

7. Laird, J.E., Newell, A. & Rosenbloom, P.S. "Soar: An architecture for general intelligence.". *Artificial Intelligence 33* (1987), 1-64.

8. Lewis, C.L. "Why and how to learn why: Analysis-based generalization of procedures". *Cognitive Science 12* (1988), 211-256.

9. Mitchell, T. M., Keller, R. & Kedar-Cabelli, S. "Explanation-Based Generalization: A Unifying View". *Machine Learning 1* (1986), 47-80.

10. Murphy, G.L. & Medin, D.L. "The role of theories in conceptual coherence.". *Psychological Review 92* (1985), 289-316.

11. Piaget, J.. *Structuralism.* Basic Books, New York, NY., 1970.

12. Rosenbloom, P.S. & Laird, J.E. Mapping explanation-based generalization into Soar. Proceedings of Fifth National Conference on Artificial Intelligence, AAAI, 1986, pp. 561-567. Philadelphia.

13. A.M. Segre (Ed.). *Proceedings of Sixth International Machine Learning Workshop.* Morgan Kaufman, San Mateo, CA, 1989. Ithaca, NY.

14 Shultz, T.R. "Rules of causal attribution". *Monographs of the Society for Research in Child Development 47* (1982), 1-51.

15. Siegler, R.S. "Developmental sequences within and between concepts". *Monographs of the Society for Research in Child Development 46* (1981), 1-74.

16. Simon, T. Analysis-based concept formation as a mechanism of cognitive development. in preparation.

17. Vadhan, V. "Induction of conservation by discrimination training". *The Journal of Psychology 116* (1984), 273-277.